A Sparse Johnson-Lindenstrauss Transform Using Fast Hashing

Jakob Bæk Tejs Houen
BARC, Department of Computer Science, University of Copenhagen, Denmark

Mikkel Thorup
BARC, Department of Computer Science, University of Copenhagen, Denmark

Abstract

The *Sparse Johnson-Lindenstrauss Transform* of Kane and Nelson (SODA 2012) provides a linear dimensionality-reducing map $A \in \mathbb{R}^{m \times u}$ in ℓ_2 that preserves distances up to distortion of $1 + \varepsilon$ with probability $1 - \delta$, where $m = O(\varepsilon^{-2} \log 1/\delta)$ and each column of A has $O(\varepsilon m)$ non-zero entries. The previous analyses of the Sparse Johnson-Lindenstrauss Transform all assumed access to a $\Omega(\log 1/\delta)$-wise independent hash function. The main contribution of this paper is a more general analysis of the Sparse Johnson-Lindenstrauss Transform with less assumptions on the hash function. We also show that the *Mixed Tabulation hash function* of Dahlgaard, Knudsen, Rotenberg, and Thorup (FOCS 2015) satisfies the conditions of our analysis, thus giving us the first analysis of a Sparse Johnson-Lindenstrauss Transform that works with a practical hash function.

2012 ACM Subject Classification Theory of computation → Random projections and metric embeddings; Theory of computation → Pseudorandomness and derandomization

Keywords and phrases dimensionality reduction, hashing, concentration bounds, moment bounds

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.76

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2305.03110

Funding Research supported by Investigator Grant 16582, Basic Algorithms Research Copenhagen (BARC), from the VILLUM Foundation.

1 Introduction

Dimensionality reduction is an often applied technique to obtain a speedup when working with high dimensional data. The basic idea is to map a set of points $X \subseteq \mathbb{R}^u$ to a lower dimension while approximately preserving the geometry. The Johnson-Lindenstrauss lemma [24] is a foundational result in that regard.

▶ **Lemma 1** ([24]). *For any $0 < \varepsilon < 1$, integers n, u, and $X \subseteq \mathbb{R}^u$ with $|X| = n$, there exists a map $f \colon X \to \mathbb{R}^m$ with $m = O(\varepsilon^{-2} \log n)$ such that*

$$\forall w, w' \in X,\ \left| \|f(w) - f(w')\|_2 - \|w - w'\|_2 \right| \le \varepsilon \|w - w'\|_2 \,.$$

It has been shown in [6, 30] that the target dimension m is optimal for nearly the entire range of n, u, ε. More precisely, for any n, u, ε there exists a set of points $X \subseteq \mathbb{R}^u$ with $|X| = n$ such that for any map $f \colon X \to \mathbb{R}^m$ where the Euclidean norm is distorted by at most $(1 \pm \varepsilon)$ must have $m = \Omega(\min\{u, n, \varepsilon^{-2} \log(\varepsilon^2 n)\})$.

50th International Colloquium on Automata, Languages, and Programming (ICALP 2023).
Editors: Kousha Etessami, Uriel Feige, and Gabriele Puppis; Article No. 76; pp. 76:1–76:20
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

All known proofs of the Johnson-Lindenstrauss lemma constructs a linear map f. The original proof of Johnson and Lindenstrauss [24] chose $f(x) = \Pi x$ where $\Pi \in \mathbb{R}^{m\times u}$ is an appropriately scaled orthogonal projection into a random m-dimensional subspace. Another simple construction is to set $f(x) = \frac{1}{\sqrt{m}}Ax$ where $A \in \mathbb{R}^{m\times u}$ and each entry is an independent Rademacher variable.[1] In both cases, it can be shown that as long as $m = \Omega(\varepsilon^{-2}\log 1/\delta)$ then

$$\forall w \in \mathbb{R}^u,\ \Pr\left[\left|\|f(w)\|_2^2 - \|w\|_2^2\right| \geq \varepsilon\|w\|_2^2\right] \leq \delta. \tag{1}$$

The Johnson-Lindenstrauss lemma follows by setting $\delta < 1/\binom{n}{2}$ and taking $w = z - z'$ for all pairs $z, z' \in X$ together with a union bound. (1) is also known as the distributional Johnson-Lindenstrauss lemma and it has been shown that the target dimension m is tight, more precisely, m must be at least $\Omega(\min\left\{u, \varepsilon^{-2}\log 1/\delta\right\})$ [23, 26].

Sparse Johnson-Lindenstrauss Transform

One way to speed up the embedding time is replacing the dense A of the above construction by a sparse matrix. The first progress in that regard came by Achlioptas in [3] who showed that A can be chosen with i.i.d. entries where $A_{ij} = 0$ with probability $2/3$ and otherwise A_{ij} is chosen uniformly in $\pm\sqrt{\frac{3}{m}}$. He showed that this construction can achieve the same m as the best analyses of the Johnson-Lindenstrauss lemma. Hence this achieves essentially a 3x speedup, but the asymptotic embedding time is still $O(m\|x\|_0)$ where $\|x\|_0$ is number of non-zeros of x.

Motivated by improving the asymptotic embedding time, Kane and Nelson in [28], following the work in [14, 27, 8], introduced the Sparse Johnson-Lindenstrauss Transform which maps down to essentially optimal dimension $m = O(\varepsilon^{-2}\log n)$ and only has $s = O(\varepsilon^{-1}\log n)$ non-zeros entries per column. This speeds up the embedding time to $O(\varepsilon^{-1}\log n\|x\|_0) = O(\varepsilon m\|x\|_0)$ thus improving the embedding time by a factor of ε^{-1}. It nearly matches a sparsity lower bound by Nelson and Nguyen [31] who showed that any sparse matrix needs at least $s = \Omega(\varepsilon^{-1}\log(n)/\log(1/\varepsilon))$ non-zeros per column.

Using Hashing

When the input dimension, u, is large it is not feasible to store the matrix A explicitly. Instead, we use a hash function to calculate the non-zero entries of A. Unfortunately, the previous analyses of the Sparse Johnson-Lindenstrauss Transform [28, 10] assume access to a $\Omega(\log 1/\delta)$-wise independent hash function which is inefficient. This motivates the natural question:

What are the sufficient properties we need of the hash function for a Sparse Johnson-Lindenstrauss Transform to work?

The goal of this work is to make progress on this question. In particular, we provide a new analysis of a Sparse Johnson-Lindenstrauss Transform with fewer assumptions on the hash function. This improved analysis allows us to conclude that there exists a Sparse Johnson-Lindenstrauss Transform that uses Mixed Tabulation hashing which is efficient.

[1] A Rademacher variables, X, is a random variable that is chosen uniformly in ± 1, i.e., $\Pr[X = 1] = \Pr[X = -1] = \frac{1}{2}$.

Mixed Tabulation Hashing

Before introducing Mixed Tabulation hashing, we will first discuss *Simple Tabulation hashing* which was introduced by Zobrist [39]. Simple Tabulation hashing takes an integer parameter $c > 1$, and we view a key $x \in [u] = \{0, \dots, u-1\}$ as a vector of c characters, $x_0, \dots, x_{c-1} \in \Sigma = [u^{1/c}]$. For each character, we initialize a fully random table $T_i \colon \Sigma \to [2^r]$ and the hash value of x is then calculated as

$$h(x) = T_0[x_0] \oplus \ldots \oplus T_{c-1}[x_{c-1}],$$

where $\oplus$ is the bitwise XOR-operation. We say that h is a Simple Tabulation hash function with c characters.

We can now define *Mixed Tabulation hashing* which is a variant of Simple Tabulation hashing that was introduced in [11]. As with Simple Tabulation hashing, Mixed Tabulation hashing takes $c > 1$ as a parameter, and it takes a further integer parameter $d \geq 1$. Again, we view a key $x \in [u]$ as vector of c characters, $x_0, \dots, x_{c-1} \in \Sigma = [u^{1/c}]$. We then let $h_1 \colon \Sigma^c \to [2^r]$, $h_2 \colon \Sigma^c \to \Sigma^d$, and $h_3 \colon \Sigma^d \to [2^r]$ be independent Simple Tabulation hashing. Mixed Tabulation hashing is then defined as follows

$$h(x) = h_1(x) \oplus h_3(h_2(x)).$$

We say that h a mixed tabulation hash function with c characters and d derived characters. We call $h_2(x) \in \Sigma^d$ the *derived* characters. Mixed Tabulation hashing can be efficiently implemented by storing h_1 and h_2 as a single table with entries in $[2^r] \times \Sigma^d$, so the whole hash function can be computed with just $c + d$ lookups.

Our Contributions

Our main contribution is a new analysis of a Sparse Johnson-Lindenstrauss Transform that does not rely on the high independence of the hash function. Instead we show that it suffices that the hash function supports a decoupling-decomposition combined with strong concentration bounds.

We show that Mixed Tabulation hashing satisfies these conditions. This gives the first instance of a practical hash function that can support a Sparse Johnson-Lindenstrauss Transform.

1.1 Sparse Johnson-Lindenstrauss Transform

As mentioned earlier, the Sparse Johnson-Lindenstrauss Transform was introduced by Kane and Nelson [28] and they provided two different constructions with the same sparsity. Later a simpler analysis was given in [10] which also generalized the result to a more general class of constructions. In this paper, we will only focus on one of the constructions which is described below.

Before we discuss the construction of the Sparse Johnson-Lindenstrauss Transform, we will first consider the related CountSketch which was introduced in [9] and was analyzed for dimensionality reduction in [36]. In CountSketch, we construct the matrix A as follows: We pick a pairwise independent hash function, $h \colon [u] \to [m]$, and a 4-wise independent sign function $\sigma \colon [u] \to \{-1, 1\}$. For each $x \in [u]$, we set $A_{h(x),x} = \sigma(x)$ and the rest of the x'th column to 0. Clearly, this construction has exactly 1 non-zero entry per column. It was shown in [36] that if $m = \Omega(\varepsilon^{-2}\delta^{-1})$ then it satisfies the distributional Johnson-Lindenstrauss lemma, Equation (1). The result follows by bounding the second moment of $\|Ax\|_2^2 - \|x\|_2^2$ for any $x \in \mathbb{R}^d$ and then apply Chebyshev's inequality.

The bad dependence in the target dimension, m, on the failure probability, δ, is because we only use the second moment. So one might hope that you can improve the dependence by looking at higher moments instead. Unfortunately, it is not possible to improve the dependence for general $x \in \mathbb{R}^d$, and it is only possible to improve the dependence if $\|x\|_\infty^2 / \|x\|_2^2$ is small. Precisely, how small $\|x\|_\infty^2 / \|x\|_2^2$ has to be, has been shown in [17]. So to improve the dependence on δ, we need to increase the number of non-zero entries per column.

We are now ready to describe the construction of the Sparse Johnson-Lindenstrauss Transform. The construction is to concatenate s CountSketch matrices and scale the resulting matrix by $\frac{1}{\sqrt{s}}$. This clearly gives a construction that has s non-zero entries per column and as it has been shown in [28, 10] if $s = \Omega(\varepsilon^{-1}\log(1/\delta))$ then we can obtain the optimal target dimension $m = O(\varepsilon^{-2}\log(1/\delta))$. More formally, we construct the matrix A as follows:

1. We pick a hash function, $h\colon [s] \times [u] \to [m/s]$ and a sign function $\sigma\colon [s] \times [u] \to \{-1, 1\}$.
2. For each $x \in [u]$, we set $A_{i \cdot m/s + h(i,x), x} = \frac{\sigma(i,x)}{\sqrt{s}}$ for every $i \in [s]$ and the rest of the x'th column to 0.

In the previous analyses [28, 10], it was shown that if h and σ are $\Omega(\log 1/\delta)$-wise independent then the construction works. Unfortunately, it is not practical to use a $\Omega(\log 1/\delta)$-wise independent hash function so the goal of this work is to obtain an analysis of a Sparse Johnson-Lindenstrauss Transform with fewer assumptions about the hash function. In particular, we relax the assumptions of the hash function, h, and the sign function, σ, to just satisfying a decoupling-decomposition and a strong concentration property. The formal theorem is stated in Section 3.

We also show that Mixed Tabulation satisfies these properties and thus that the Sparse Johnson-Lindenstrauss Transform can be implemented using Mixed Tabulation. Let us describe more formally, what we mean by saying that Mixed Tabulation can implement the Sparse Johnson-Lindenstrauss Transform. We let $h_1\colon \Sigma^c = [u] \to [m/s]$, $h_2\colon \Sigma^c \to \Sigma^d$, and $h_3\colon \Sigma^d \to [m/s]$ be the independent Simple Tabulation hash functions that implement the Mixed Tabulation hash function, $h_1(x) \oplus h_3(h_2(x))$. We then extend it to the domain $[s] \times [u]$ as follows:

1. Let $h'_2\colon [s] \times \Sigma^c \to \Sigma^d$ be defined by $h'_2(i,x) = h_2(x) \oplus \underbrace{(i, \ldots, i)}_{d \text{ times}}$, i.e., each derived character gets xor'ed by i.
2. We then define $h\colon [s] \times [u] \to [m/s]$ and $\sigma\colon [s] \times [u] \to \{-1, 1\}$ by $h(i,x) = h_1(x) \oplus h_3(h'_2(i,x))$ and $\sigma(i,x) = \sigma_1(x) \cdot \sigma_3(h'_2(i,x))$, where h_1 and h_3 are the Simple Tabulation hash functions described above, and $\sigma_1\colon \Sigma^c \to \{-1, 1\}$ and $\sigma_3\colon \Sigma^d \to \{-1, 1\}$ are independent Simple Tabulation functions.

1.2 Hashing Speed

When we use tabulation schemes, it is often as a fast alternative to $\Omega(\log n)$-independent hashing. Typically, we implement a q-independent hash function using a degree $q - 1$ polynomial in $O(q)$ time, and Siegel [34] has proved that this is best possible unless we use large space. More precisely, for some key domain $[u]$, if we want to do $t < q$ memory accesses, then we need space at least $u^{1/t}$. Thus, if we want higher than constant independence but still constant evaluation times, then we do need space $u^{\Omega(1)}$. In our application, we have to compute many hash values simultaneously, so an alternative strategy would be to evaluate the polynomial using multi-point evaluation. This would reduce the time per hash value to $O(\log^2 q)$ but this is still super constant time.

With tabulation hashing, we use tables of size $O(|\Sigma|)$ where $|\Sigma| = u^{1/c}$ and $c = O(1)$. The table lookups are fast if the tables fit in cache, which is easily the case for 8-bit characters. In connection with each lookup, we do a small number of very fast AC^0 operations: a cast, a bit-wise xor, and a shift. This is incomparable to polynomial in the sense of fast cache versus multiplications, but the experiments from [1, Table 1] found Simple Tabulation hashing to be faster than evaluation a 2-wise independent polynomial hashing.

Tabulation schemes are most easily compared by the number of lookups. Storing h_1 and h_2 in the same table, Mixed Tabulation hashing uses $c + d$ lookups. With $d = c$, the experiments from [1] found Mixed Tabulation hashing to be slightly more than twice as slow as Simple Tabulation hashing, and the experiments from [12] found Mixed Tabulation hashing to be about as fast as 3-wise independent polynomial hashing. This motivates our claim that Mixed Tabulation hashing is practical.

In theory, we could also use a highly independent hash function that uses large space, but we don't know of any efficient construction. Siegel states about his construction, it is "far too slow for any practical application" [34], and while Thorup [35] has presented a simpler construction than Siegel's, it is still not efficient. The experiments in [1] found it to be more than an order magnitude slower than Mixed Tabulation hashing.

2 Related Work

Even Sparser Johnson-Lindenstrauss Transforms

As touched upon earlier, there is a lower bound by Nelson and Nguyen [31] that rules out significant improvements, but never the less there has been research into sparser embedding. In the extreme, Feature Hashing of [38] considers the case of $s = 1$. The lower bound excludes Feature Hashing from working for all vectors, but in [17] they gave tight bounds for which vectors it works in terms of the measure $\|w\|_\infty^2 / \|w\|_2^2$. This was later generalized in [21] to a complete understanding between the tradeoff between s and the measure $\|w\|_\infty^2 / \|w\|_2^2$. In this paper, we will only focus on the case $s = \Theta(\varepsilon^{-1} \log 1/\delta)$ and $m = \Theta(\varepsilon^{-2} \log 1/\delta)$

Fast Johnson-Lindenstrauss Transform

Another direction to speed-up the evaluation of Johnson-Lindenstrauss transforms is to exploit dense matrices with fast matrix-vector multiplication. This was first done by Ailon and Chazelle [4] who introduced the Fast Johnson-Lindenstrauss Transform. Their original construction was recently [16] shown to give an embedding time $O(u \log u + m(\log 1/\delta + \varepsilon \log^2(1/\delta)/\log(1/\varepsilon)))$.

This has generated a lot follow-up work that has tried to improve the running to a clean $O(u \log u)$. Some of the work sacrifice the optimal target dimension, $m = O(\varepsilon^{-2} \log 1/\delta)$, in order to speed-up the construction, and are satisfied with sub-optimal $m = O(\varepsilon^{-2} \log n \log^4 u)$ [29], $m = O(\varepsilon^{-2} \log^3 n)$ [15], $m = O(\varepsilon^{-1} \log^{3/2} n \log^{3/2} u + \varepsilon^{-2} \log n \log^4 u)$ [29], $m = O(\varepsilon^{-2} \log^2 n)$ [19, 37, 18], and $m = O(\varepsilon^{-2} \log n \log^2(\log n) \log^3 u)$ [22]. Another line of progress is to assume that the target dimension, m, is substantially smaller then the starting dimension, u. Under the assumption that $m = o(u^{1/2})$ the work in [5, 7] achieves embedding time $O(u \log m)$. The only construction that for some regimes improves on the original Fast Johnson-Lindenstrauss Transform is the recent analysis [22] of the Kac Johnson-Lindenstrauss Transform, which uses the Kac random walk [25]. They show that it can achieve an embedding time of $O(u \log u + \min\{u \log n, m \log n \log^2(\log n) \log^3 u\})$.

Previous Work on Tabulation Hashing

The work by Patrascu and Thorup [33] initiated the study of tabulation based hashing that goes further than what 3-wise independence of constructions would suggest. A long line of papers have shown tabulation based hashing to work for min-wise hashing [32, 13], hashing for k-statistics [11], and the number of non-empty-bins [2]. Furthermore, multiple papers have been concerned with showing strong concentration results for tabulation based hashing [33, 32, 1, 20]. Tabulation based hashing has also been studied experimentally where they have been shown to exhibit great performance [12, 1].

Preliminaries

In this section, we will introduce the notation which will be used throughout the paper. First we introduce p-norms.

▶ **Definition 2** (p-norm)**.** *Let $p \geq 1$ and X be a random variable with $\mathrm{E}[|X|^p] < \infty$. We then define the p-norm of X by $\|X\|_p = \mathrm{E}[|X|^p]^{1/p}$.*

Throughout the paper, we will repeatedly work with value functions $v \colon U \times [m] \to \mathbb{R}$. We will allow ourself to sometime view them as vectors, and in particular, we will write

$$\|v\|_2 = \sqrt{\sum_{x \in U} \sum_{j \in [m/s]} v(x,j)^2},$$
$$\|v\|_\infty = \max_{x \in U, j \in [m/s]} |v(x,j)| \, .$$

We will also use the Ψ_p-function introduced in [20].

▶ **Definition 3.** *For $p \geq 2$ we define the function $\Psi_p \colon \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ as follows,*

$$\Psi_p(M, \sigma^2) = \begin{cases} \left(\frac{\sigma^2}{pM^2}\right)^{1/p} M & \text{if } p < \log \frac{pM^2}{\sigma^2} \\ \frac{1}{2}\sqrt{p}\sigma & \text{if } p < e^2 \frac{\sigma^2}{M^2} \\ \frac{p}{e \log \frac{pM^2}{\sigma^2}} M & \text{if } \max\left\{\log \frac{pM^2}{\sigma^2}, e^2 \frac{\sigma^2}{M^2}\right\} \leq p \end{cases} .$$

It was shown in [20] that $\Psi_p(1, \lambda)$ is within a constant factor of the p-norm of a Poisson distributed random variable with parameter λ. They also showed that $\Psi_p(M, \sigma^2)$ can be used to upper bound expressions involving a fully random hash function $h \colon U \to [m]$. Let $v \colon U \times [m] \to \mathbb{R}$ be a value function then they showed that

$$\left\| \sum_{x \in U} v(x, h(x)) \right\|_{\leq} C \Psi_p(\|v\|_\infty, \|v\|_2^2 / m) \, ,$$

where C is a universal constant.

3 Overview of the New Analysis

Our main technical contribution is a new analysis of the Sparse Johnson-Lindenstrauss Transform that relaxes the assumptions on the hash function, h. We show that if h satisfies a decoupling decomposition property and a strong concentration property then we obtain the same bounds for the Sparse Johnson-Lindenstrauss Transform. Both of these properties are satisfied by h if h is $\Omega(\log 1/\delta)$-wise independent so our assumptions are weaker than those of the previous analyses.

In this section, we will give an informal overview of new approach. The technical details and the formal statement of the result will be in Section 4.

In order to describe our approach, we look at the random variable

$$Z = \|Aw\|_2^2 - 1 = \frac{1}{s} \sum_{i \in [s]} \sum_{x \neq y \in [u]} \sigma(i,x)\sigma(i,y)\,[h(i,x) = h(i,y)]\,w_x w_y. \tag{2}$$

Here $w \in \mathbb{R}^u$ is a unit vector. With this notation the goal becomes to bound $\Pr[|Z| \geq \varepsilon]$.

The first step in our analysis is that we want to decouple Equation (2). Decoupling was also used in one of the proofs in [10], but since we want to prove the result for more general hash functions, we cannot directly use the standard decoupling inequalities. We will instead assume that our hash function allows a *decoupling-decomposition.* This will formally be defined in Section 4 and we will for now assume that our hash function allows for the standard decoupling inequality. If we apply Markov's inequality and a standard decoupling inequality for fully random hashing we obtain the expression.

$$\begin{aligned}\Pr[|Z| \geq \varepsilon] &\leq \varepsilon^{-p}\,\mathrm{E}[|Z|^p] \\ &\leq \left(\varepsilon^{-1}\frac{4}{s}\right)^p \mathrm{E}\left[\left|\sum_{i\in[s]}\sum_{x,y\in[u]} \sigma(i,x)\sigma'(i,y)\,[h(i,x) = h'(i,y)]\,w_x w_y\right|^p\right]\end{aligned} \tag{3}$$

where (h', σ') are independent copies of (h, σ) and $p \geq 2$. The power of decoupling stems from the fact that it breaks up some of the dependencies and allows for a simpler analysis.

The goal is now to analyse $\left\|\sum_{i\in[s]}\sum_{x,y\in[u]} \sigma(i,x)\sigma'(i,y)\,[h(i,x) = h'(i,y)]\,w_x w_y\right\|_p$. This is done by first fixing (h', σ') and bounding $\left\|\sum_{i\in[s],j\in[m/s]}\sum_{x\in[u]} \sigma(i,x)\,[h(i,x) = j]\,w_x a_{ij}\right\|_p$ using the randomness of (h, σ) where $a_{ij} = \sum_{y\in[u]} \sigma'(i,y)\,[h'(i,y) = j]\,w_y$. In order to do this, we will assume that the pair (h, σ) is *strongly concentrated.* Again the formal definition of this is postponed to Section 4, but informally, we say that the pair is strongly concentrated if it has concentration results similar to those of fully random hashing.

We now take the view that $|a_{ij}|$ is the load of the bin $(i,j) \in [s] \times [m/s]$. The idea is then to split $[s] \times [m/s]$ into heavy and light bins and handle each separately. We choose a parameter k and let I be the heaviest k bins. Using the triangle inequality, we then get that

$$\begin{aligned}\left\|\sum_{i\in[s],j\in[m/s]}\sum_{x\in[u]} \sigma(i,x)\,[h(i,x) = j]\,w_x a_{ij}\right\|_p &\leq \left\|\sum_{(i,j)\in I}\sum_{x\in[u]} \sigma(i,x)\,[h(i,x) = j]\,w_x a_{ij}\right\|_p \\ &+ \left\|\sum_{(i,j)\in[s]\times[m/s]\setminus I}\sum_{x\in[u]} \sigma(i,x)\,[h(i,x) = j]\,w_x a_{ij}\right\|_p .\end{aligned}$$

We show that the contribution from the light bins is as if the collisions are independent. This should be somewhat intuitive since if we only have few collisions in each bin then the collisions behave as if they were independent. In contrast, we show that the contribution from the heavy bins is dominated by the heaviest bin. This turns out to be exactly what we need to finish the analysis.

4 Technical Results

In this section, we will expand on the description from Section 3 and formalize the ideas.

Decoupling

Ideally, we would like to use the standard decoupling inequality, Equation (3). Unfortunately, we cannot expect more general hash functions to support such a clean decoupling. We therefore introduce the notion of a decoupling-decomposition.

▶ **Definition 4** (Decoupling-decomposition). *Let $p \geq 2$, $L \geq 1$, and $0 \leq \gamma \leq 1$. We say that a collection of possibly randomized sets, (U_α), is a (p, L, γ)*-decoupling-decomposition *for a property P of a pair (h, σ), if there exist hash functions $h_\alpha \colon [s] \times U_\alpha \to [m/s]$ and sign functions $s \colon [s] \times U_\alpha \to \{-1, 1\}$ for all α such that*

$$\Pr[|Z| \geq \varepsilon] \\ \leq \left(\varepsilon^{-1} \sum_\alpha \frac{L}{s} \left\| \sum_{i \in [s]} \sum_{x, y \in U_\alpha} \sigma_\alpha(i, x) \sigma'_\alpha(i, y) \, [h_\alpha(i, x) = h'_\alpha(i, y)] \, w_x w_y \right\|_p \right)^p + \gamma \tag{4}$$

where $(h_\alpha, \sigma_\alpha)$ and $(h'_\alpha, \sigma'_\alpha)$ has the same distribution, and $(h_\alpha, \sigma_\alpha)$ satisfies the property P when conditioned on $(h'_\alpha, \sigma'_\alpha)$ and U_α.

The reader should compare Equation (3) for fully random hashing with Equation (4). There are 3 main differences between the expressions.

1. The first thing to notice is that, in the decoupling-decomposition we sum over different sets (U_α), where this is not needed for fully random hashing. We allow the decoupling-decomposition to use a different decoupling on each of the sets U_α. This is very powerful since general hash functions are not necessarily uniform over the input domain.
2. For the decoupling-decomposition, we allow an additive error probability γ. This is useful if the hash function allows for decoupling most of the time except when some unprobable event is happening.
3. The last difference is that a much larger loss-factor is allowed by the decoupling-decomposition than Equation (3). In the case of fully random hashing, we only lose a factor of 4 but for more general hash functions this loss might be bigger.

Finally, we note that Equation (3) implies if (h, σ) is $2p$-wise independent for an integer $p \geq 2$ then $[u]$ is a decoupling-decomposition of (h, σ) for any property P that is satisfied by (h, σ).

Strong Concentration

The second property we need is that the hash function is strongly concentrated.

▶ **Definition 5** (Strong concentration). *Let $h \colon [s] \times U \to [m/s]$ be a hash function and $\sigma \colon [s] \times U \to \{-1, 1\}$ be a sign function. We say that the pair (h, σ) is (p, L)*-strongly-concentrated *if*

1. *For all value functions, $v\colon [s] \times [m/s] \to \mathbb{R}$, and all vectors, $w \in \mathbb{R}^U$,*

$$\left\| \sum_{i \in [s]} \sum_{x \in U} \sigma(i,x) v(i, h(i,x)) w_x \right\|_p \le \Psi_p\left(L \|v\|_\infty \|w\|_\infty , L \frac{s}{m} \|v\|_2^2 \|w\|_2^2 \right) , \tag{5}$$

$$\left\| \sum_{i \in [s]} \sum_{x \in U} \sigma(i,x) v(i, h(i,x)) w_x \right\|_p \le \sqrt{L \frac{p}{\log(m/s)} \|v\|_2^2 \|w\|_2^2} . \tag{6}$$

2. *For all vectors, $w \in \mathbb{R}^U$,*

$$\left\| \sum_{i \in [s]} \sum_{j \in [m/s]} \left(\sum_{x \in U} \sigma(i,x) \left[h(i,x) = j \right] w_x \right)^2 \right\|_{p/2} \le L \max\left\{ s \|w\|_2^2 , \frac{p}{\log m/s} \|w\|_2^2 \right\} . \tag{7}$$

3. *If $p \le \log m$,*

$$\left\| \max_{i \in [s], j \in [m/s]} \left| \sum_{x \in U} \sigma(i,x) \left[h(i,x) = j \right] w_x \right| \right\|_p \le e \sqrt{L \frac{\log m}{\log m/s}} \|w\|_2 . \tag{8}$$

We need essentially 3 different properties of our hash function to say that it is strongly concentrated.

1. The first property is a concentration result on the random variable

$$\sum_{i \in [s]} \sum_{x \in U} \sigma(i,x) v(i, h(i,x)) w_x .$$

   Here we need two different concentration results: The first concentration result, Equation (5), roughly corresponds to a p-norm version of what you would obtain by applying Bennett's inequality to a fully random hash function, while the second concentration result, Equation (5), corresponds to the best hypercontractive result you can obtain for weighted sums of independent Bernoulli-Rademacher variables with parameter s/m.[2]

2. The second property bounds the sum of squares

$$W = \sum_{i \in [s]} \sum_{j \in [m/s]} \left(\sum_{x \in U} \sigma(i,x) \left[h(i,x) = j \right] w_x \right)^2 .$$

   The condition, Equation (7), bounds $\|W\|_{p/2}$ by the maximum of two cases. The first case corresponds to $\mathrm{E}[W]$, and the second case is motivated by applying Equation (6) to

$$\sup_{\substack{z \in \mathbb{R}^{[s] \times [m/s]}, \\ \|z\|_2 = 1}} \left\| \sum_{i \in [s]} \sum_{j \in [m]} \sum_{x \in U} \sigma(i,x) z_{i, h(i,x)} w_x \right\|_p^2 .$$

   While this at first glance might seem odd, it is roughly the best you can do, since one can show that

$$\max\left\{ \mathrm{E}[W] , \sup_{\substack{z \in \mathbb{R}^{[s] \times [m/s]}, \\ \|z\|_2 = 1}} \left\| \sum_{i \in [s]} \sum_{j \in [m]} \sum_{x \in U} \sigma(i,x) z_{i, h(i,x)} w_x \right\|_p^2 \right\} \le \|W\|_{p/2} .$$

[2] A Bernoulli-Rademacher variable with parameter α is random variable, $X \in \{-1, 0, 1\}$, with $\Pr[X = 1] = \Pr[X = -1] = \alpha/2$ and $\Pr[X = 0] = 1 - \alpha$.

3. The final property is a bound on the largest coordinate, $\max_{i \in [s], j \in [m/s]} \left| \sum_{x \in U} \sigma(i,x) \left[h(i,x) = j\right] w_x \right|$. The bound is a natural consequence of Equation (6) for fully random hashing. Namely, for fully random hashing we get that

$$\begin{aligned}
&\left\| \max_{i \in [s], j \in [m/s]} \left| \sum_{x \in U} \sigma(i,x) \left[h(i,x) = j\right] w_x \right| \right\|_p \\
&\quad \le \left\| \max_{i \in [s], j \in [m/s]} \left| \sum_{x \in U} \sigma(i,x) \left[h(i,x) = j\right] w_x \right| \right\|_{\log m} \\
&\quad \le e \max_{i \in [s], j \in [m/s]} \left\| \left| \sum_{x \in U} \sigma(i,x) \left[h(i,x) = j\right] w_x \right| \right\|_{\log m} \\
&\quad \le e \sqrt{L \frac{\log m}{\log m/s}} \left\| w \right\|_2 .
\end{aligned}$$

This derivation is not true for general hash function, but the hash function can still satisfy Equation (8).

The results of [20] show that if the hash function $h \colon [s] \times U \to [m/s]$ and the sign function $\sigma \colon [s] \times U \to [m/s]$ is p-wise independent for an integer $p \ge 2$ then the pair (h, σ) is (p, K)-strongly-concentrated where K is a universal constant.

The Main Result

We are now ready to state our main result which is a new analysis of a Sparse Johnson-Lindenstrauss Transform that only assumes that the hash function has a decoupling-decomposition for the strong concentration property.

▶ **Theorem 6.** *Let $h \colon [s] \times [u] \to [m/s]$ be a hash function and $\sigma \colon [s] \times [u] \to \{-1, 1\}$ be a sign function. Furthermore, let $0 < \varepsilon < 1$ and $0 < \delta < 1$ be given, and define $p = \log 1/\delta$.*

Assume that there exists constants L_1, L_2, L_3, and $0 \le \gamma < 1$, that only depends on (h, σ) and p, such that

1. *There exists a (p, L_1, γ)-decoupling-decomposition, (U_α), for the (p, L_2)-strong-concentration property of (h, σ)*
2. *For all vectors $w \in \mathbb{R}^u$, $\sum_\alpha \sum_{x \in U_\alpha} w_x^2 \le L_3 \left\| w \right\|_2^2$.*
3. *$m \ge \left(16 e^7 L_1^2 L_2^3 L_3^2\right) \cdot \varepsilon^{-2} \log(1/\delta)$.*
4. *$s \ge \left(64 e^3 L_1 L_2^{3/2} L_3\right) \cdot \varepsilon^{-1} \log(1/\delta)$.*

Then the following is true

$\Pr[|Z| \ge \varepsilon] \le \delta + \gamma.$

As discussed earlier, a fully random hash function satisfies all the property needed of the theorem and thus gives a new analysis of the Sparse Johnson-Lindenstrauss Transform for fully random hashing. We will also later show that Mixed Tabulation satisfies the assumption of the theorem hence giving the first analysis of a Sparse Johnson-Lindenstrauss Transform with a practical hash function that works.

The main difficulty in the analysis of Theorem 6 is contained in the following technical lemma. The idea in the proof of Theorem 6 is to use the decoupling-decomposition and apply the following lemma to each part.

▶ **Lemma 7.** *Let $h, \overline{h}\colon [s] \times U \to [m/s]$ be hash functions and $\sigma, \overline{\sigma}\colon [s] \times U \to \{-1, 1\}$ be sign functions. Let $p \geq 2$ and assume that there exists a constant L such that (h, σ) is (p, L)-strongly concentrated when conditioning on $(\overline{h}, \overline{\sigma})$, and similarly, $(\overline{h}, \overline{\sigma})$ is (p, L)-strongly concentrated when conditioning on (h, σ). Then for all vectors $w \in \mathbb{R}^U$,*

$$\left\| \sum_{i \in [s]} \sum_{x, y \in U} \sigma(i,x)\overline{\sigma}(i,y) \left[h(i,x) = \overline{h}(i,y)\right] w_x w_y \right\|_p \\ \leq \Psi_p\left(32e^3 L^{3/2} \|w\|_2^2, 32e^6 L^3 \frac{s^2}{m} \|w\|_2^4\right) + 36e^3 L \frac{p}{\log m/s} \|w\|_2^2 .$$

The lemma shows that the expression has two different regimes. The first regime, $\Psi_p\left(32e^3 L^{3/2} \|w\|_2^2, 32e^6 L^3 \frac{s^2}{m} \|w\|_2^4\right)$, is essentially what we would expect if each of the collisions, $\left[h(i,x) = \overline{h}(i,y)\right]$, are independent of each other. The other regime, $36e^3 L \frac{p}{\log m/s} \|w\|_2^2$, is essentially what you expect the largest coordinate to contribute.

Our analysis is inspired by these two regimes and tries to exploit them explicitly. We start by fixing (h, σ) and divide the coordinates into heavy and light coordinates. We then show that contribution of the light coordinates is $\Psi_p\left(32e^3 L^{3/2} \|w\|_2^2, 32e^6 L^3 \frac{s^2}{m} \|w\|_2^4\right)$ which matches the intuition that if we only have few collisions on each coordinate then the collisions behave as if they were independent. Similarly, we show that the contribution of the heavy coordinates is dominated by the heaviest coordinate, namely, the contribution is $36e^3 L \frac{p}{\log m/s} \|w\|_2^2$.

Mixed Tabulation Hashing

Our main result for Mixed Tabulation hashing is the following.

▶ **Theorem 8.** *Let $h\colon [s] \times [u] \to [m/s]$ and $\sigma\colon [s] \times [u] \to \{-1, 1\}$ be Mixed Tabulation functions as described in Section 1.1. Furthermore, let $0 < \varepsilon < 1$ and $0 < \delta < 1$ be given, and define $p = \log 1/\delta$.*

If $m \geq \gamma_p^{3c} \varepsilon^{-2} \log(1/\delta)$ and $s \geq \gamma_p^{3/2c} \varepsilon^{-1} \log(1/\delta)$ where $\gamma_p = Kc \max\left\{1, \frac{p}{\log|\Sigma|}\right\}$ for a universal constant K.

Then the following is true

$$\Pr[|Z| \geq \varepsilon] \leq \delta + \varepsilon 3^c |\Sigma|^{-d} .$$

The result follows by proving that Mixed Tabulation hashing has a $(p, 4^{c+2}, 4\varepsilon^{-2} 3^c \frac{s}{m} |\Sigma|^{-d})$-decoupling-decomposition and that Mixed Tabulation has the strong concentration property. The main new part is in showing the decoupling-decomposition while the analysis of the strong concentration property is modification of the analysis in [20].

Due to space constraints, the proof is deferred to the full version.

5 Analysis of the Sparse Johnson-Lindenstrauss Transform

Lets us start by showing how Lemma 7 implies our main result, Theorem 6.

Proof of Theorem 6. We start by using Equation (4) of the decoupling decomposition to get that

$$\Pr[|Z| \geq \varepsilon] \\ \leq \left(\varepsilon^{-1} \sum_{\alpha} \frac{L_1}{s} \left\| \sum_{i\in[s]} \sum_{x,y\in U_\alpha} \sigma_\alpha(i,x)\sigma'_\alpha(i,y)\,[h_\alpha(i,x) = h'_\alpha(i,y)]\, v_x v_y \right\|_p \right)^p + \gamma$$

Now we fix α and apply Lemma 7 while fixing U_α

$$\left\| \sum_{i\in[s]} \sum_{x,y\in U_\alpha} \sigma_\alpha(i,x)\sigma'_\alpha(i,y)\,[h_\alpha(i,x) = h'_\alpha(i,y)]\, v_x v_y \right\|_p \\ \leq \Psi_p\left(32e^3 L_2^{3/2}, 32e^6 L^3 \frac{s^2}{m}\right) \sum_{x\in U_\alpha} w_x^2 + 36e^3 L_2 \frac{p}{\log m/s} \sum_{x\in U_\alpha} w_x^2$$

Using this we get that

$$\sum_{\alpha} \frac{L_1}{s} \left\| \sum_{i\in[s]} \sum_{x,y\in U_\alpha} \sigma_\alpha(i,x)\sigma'_\alpha(i,y)\,[h_\alpha(i,x) = h'_\alpha(i,y)]\, v_x v_y \right\|_p \\ \leq \sum_{\alpha} \frac{L_1}{s} \left(\Psi_p\left(32e^3 L_2^{3/2}, 32e^6 L^3 \frac{s^2}{m}\right) \sum_{x\in U_\alpha} w_x^2 + 36e^3 L_2 \frac{p}{\log m/s} \sum_{x\in U_\alpha} w_x^2 \right)$$

We now use that $\sum_\alpha \sum_{x\in U_\alpha} w_x^2 \leq L_3 \|w\|_2^2$ to get that

$$\sum_{\alpha} \frac{L_1}{s} \left(\Psi_p\left(32e^3 L_2^{3/2}, 32e^6 L^3 \frac{s^2}{m}\right) \sum_{x\in U_\alpha} w_x^2 + 36e^3 L_2 \frac{p}{\log m/s} \sum_{x\in U_\alpha} w_x^2 \right) \\ \leq \frac{L_3 L_1}{s} \left(\Psi_p\left(32e^3 L_2^{3/2}, 32e^6 L^3 \frac{s^2}{m}\right) + 36e^3 L_2 \frac{p}{\log m/s} \right) \|w\|_2^2$$

It can now be checked that if m and s satisfies the stated assumptions then

$$\frac{L_3 L_1}{s} \left(\Psi_p\left(32e^3 L_2^{3/2}, 32e^6 L^3 \frac{s^2}{m}\right) + 36e^3 L_2 \frac{p}{\log m/s} \right) \|w\|_2^2 \leq e^{-1}\varepsilon$$

Combining all the facts, we get that

$$\Pr[|Z| \geq \varepsilon] \leq \left(\varepsilon^{-1}(e^{-1}\varepsilon)\ \right)^p + \gamma = \delta + \gamma.$$

This finishes the proof. ◀

The rest of the section is concerned with proving our main technical lemma, Lemma 7. First we need the following two lemmas from [20].

▶ **Lemma 9.** *Let $f\colon \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a non-negative function which is monotonically increasing in every argument, and assume that there exists positive reals $(\alpha_i)_{i\in[n]}$ and $(t_i)_{i\in[n]}$ such that for all $\lambda \geq 0$,*

$$f(\lambda^{\alpha_0} t_0, \ldots, \lambda^{\alpha_{n-1}} t_{n-1}) \leq \lambda f(t_0, \ldots, t_{n-1}) \ .$$

Let $(X_i)_{i\in[n]}$ be non-negative random variables. Then for all $p \geq 1$ we have that

$$\|f(X_0, \ldots, X_{n-1})\|_p \leq n^{1/p} \max_{i\in[n]} \left(\frac{\|X_i\|_{p/\alpha_i}}{t_i} \right)^{1/\alpha_i} f(t_0, \ldots, t_{n-1}) \ .$$

▶ **Lemma 10.** *Let $p \geq 2$, $M > 0$, and $\sigma^2 > 0$ then*

$$\frac{1}{2}\sqrt{p}\sigma \leq \Psi_p(M, \sigma^2) \leq \max\left\{ \frac{1}{2}\sqrt{p}\sigma, \frac{1}{2e} pM \right\} .$$

We are now ready to prove Lemma 7.

Proof of Lemma 7. We start by defining $v_h, v_{\bar{h}}\colon [s] \times [m/s] \to \mathbb{R}$ by,

$$v_h(i,j) = \sum_{x\in U} \sigma(i,x) w_x \left[h(i,x) = j \right] \ ,$$
$$v_{\bar{h}}(i,j) = \sum_{y\in U} \overline{\sigma}(i,y) w_y \left[\overline{h}(i,y) = j \right] \ .$$

We then want to prove that

$$\left\| \sum_{i\in[s], j\in[m/s]} v_h(i,j) v_{\bar{h}}(i,j) \right\|_p \leq \Psi_p\left(32e^3 L^{3/2} \|w\|_2^2, 32e^6 L^3 \|w\|_2^4 \right) + 4e^3 L \frac{p}{\log m/s} \|w\|_2^2 \ .$$

First we consider the case where $\frac{p}{\log m/s} \|w\|_2^2 \geq s \|w\|_2^2$. By Cauchy-Schwartz and Equation (7) we get that

$$\left\| \sum_{i\in[s], j\in[m/s]} v_h(i,j) v_{\bar{h}}(i,j) \right\|_p \leq \left\| \sum_{i\in[s], j\in[m/s]} v_h(i,j)^2 \right\|_p \leq L \frac{p}{\log m/s} \|w\|_2^2 \ .$$

We now focus on the case where $\frac{p}{\log m/s} \|w\|_2^2 < s \|w\|_2^2$. We define $\pi\colon [m] \to [s] \times [m/s]$ to be a bijection which satisfies that

$$|v_h(\pi(0))| \geq |v_h(\pi(1))| \geq \ldots \geq |v_h(\pi(m-1))| \ .$$

We note that π is a random function but we can define π such that it only depends on the randomness of h and σ. We define $k = \lfloor p/\log(m/p) \rfloor$, $I = \{\pi(i) \mid i \in [k]\}$, and the random functions $v_h', v_h''\colon [s] \times [m/s] \to \mathbb{R}$ by

$$v_h'(i,j) = v_h(i,j) \left[(i,j) \in I \right] ,$$
$$v_h''(i,j) = v_h(i,j) \left[(i,j) \notin I \right] .$$

Again we note that v_h' and v_h'' only depends on the randomness of h and σ. We can then write our expression as

$$\begin{aligned}\left\|\sum_{i\in[s],j\in[m/s]} v_h(i,j)v_{\overline{h}}(i,j)\right\|_p &= \left\|\sum_{i\in[s]}\sum_{y\in U}\overline{\sigma}(i,y)v_h(i,\overline{h}(i,y))w_y\right\|_p \\ &\leq \left\|\sum_{i\in[s]}\sum_{y\in U}\overline{\sigma}(i,y)v_h'(i,\overline{h}(i,y))w_y\right\|_p + \left\|\sum_{i\in[s]}\sum_{y\in U}\overline{\sigma}(i,y)v_h''(i,\overline{h}(i,y))w_y\right\|_p .\end{aligned}$$

We will bound each of the term separately. We start by bounding $\left\|\sum_{i\in[s]}\sum_{y\in U}\overline{\sigma}(i,y)v_h'(i,\overline{h}(i,y))w_y\right\|_p$. We fix h and σ and use Equation (6) to get that

$$\begin{aligned}\left\|\sum_{i\in[s]}\sum_{y\in U}\overline{\sigma}(i,y)v_h'(i,\overline{h}(i,y))w_y\right\|_p &\leq \left\|\sqrt{L\frac{p}{\log m/s}\left\|w\right\|_2^2\left\|v_h'\right\|_2^2}\right\|_p \\ &= \sqrt{L\frac{p}{\log m/s}}\left\|w\right\|_2\left\|\sqrt{\sum_{(i,j)\in I} v_h'(i,j)^2}\right\|_p .\end{aligned}$$

We note that $\sum_{(i,j)\in I} v_h'(i,j)^2 = \max_{J\subseteq[s]\times[m/s],|J|=k}\sum_{(i,j)\in J} v_h(i,j)^2$. We then get that

$$\begin{aligned}\left\|\sqrt{\sum_{(i,j)\in I} v_h'(i,j)^2}\right\|_p &= \left\|\sqrt{\max_{J\subseteq[s]\times[m/s],|J|=k}\sum_{(i,j)\in J} v_h(i,j)^2}\right\|_p \\ &\leq \left(\sum_{J\subseteq[s]\times[m/s],|J|=k}\left\|\sqrt{\sum_{(i,j)\in J} v_h(i,j)^2}\right\|_p^p\right)^{1/p} \\ &\leq \binom{ms}{k}^{1/p}\max_{J\subseteq[s]\times[m/s],|J|=k}\left\|\sqrt{\sum_{(i,j)\in J} v_h(i,j)^2}\right\|_p\end{aligned}$$

We use Sterling's bound and get that $\binom{ms}{k}^{1/p} \leq \left(\frac{ems}{k}\right)^{k/p} \leq \left(\frac{ems\log(ms/p)}{p}\right)^{1/\log(ms/p)} \leq e^3$. So we get that

$$\left\|\sqrt{\sum_{(i,j)\in I} v_h'(i,j)^2}\right\|_p \leq e^3\max_{J\subseteq[s]\times[m/s],|J|=k}\left\|\sqrt{\sum_{(i,j)\in J} v_h(i,j)^2}\right\|_p$$

A standard volumetric argument gives that there exists a 1/4-net, $Z \subseteq \mathbb{R}^J$, with $|Z| \leq 9^k$, such that

$$\begin{aligned}
\left\|\sqrt{\sum_{(i,j)\in J} v_h(i,j)^2}\right\|_p &= \left\|\sup_{z\in\mathbb{R}^J,\|z\|_2=1}\sum_{(i,j)\in J} z_{i,j}v_h(i,j)\right\|_p \\
&\le \left\|\sup_{z\in Z}\sum_{(i,j)\in J} z_{i,j}v_h(i,j)\right\|_p \\
&\qquad + \left\|\sup_{z\in\mathbb{R}^J,\|z\|_2=1}\sum_{(i,j)\in J} (z_{i,j}-z'_{i,j})v_h(i,j)\right\|_p \\
&\le \left\|\sup_{z\in Z}\sum_{(i,j)\in J} z_{i,j}v_h(i,j)\right\|_p \\
&\qquad + \sup_{z\in\mathbb{R}^J,\|z\|_2=1}\|z-z'\|_2 \left\|\sqrt{\sum_{(i,j)\in J} v_h(i,j)^2}\right\|_p
\end{aligned}$$

where $z'\in Z$ is the closest element to z, and as such $\|z-z'\|_2\le 1/4$. Since there are at most 9^k elements in Z then $\left\|\sup_{z\in Z}\sum_{(i,j)\in J} z_{i,j}v_h(i,j)\right\|_p \le 9\sup_{z\in Z}\left\|\sum_{(i,j)\in J} z_{i,j}v_h(i,j)\right\|_p$, where we used that $k\le p$. Collecting the fact we get that

$$\left\|\sqrt{\sum_{(i,j)\in J} v_h(i,j)^2}\right\|_p \le 36\sup_{z\in Z}\left\|\sum_{(i,j)\in J} z_{i,j}v_h(i,j)\right\|_p$$

Using this we get that

$$\begin{aligned}
&e^3 \max_{J\subseteq[s]\times[m/s],|J|=k}\left\|\sqrt{\sum_{(i,j)\in J} v_h(i,j)^2}\right\|_p \\
&\quad\le 36e^3 \max_{J\subseteq[s]\times[m/s],|J|=k}\max_{z\in Z}\left\|\sum_{(i,j)\in J} z_{i,j}v_h(i,j)\right\|_p \\
&\quad= 36e^3 \max_{J\subseteq[s]\times[m/s],|J|=k}\max_{\substack{z\in\mathbb{R}^{s\times m/s},\\ \|z\|_2=1}}\left\|\sum_{i\in[s]}\sum_{x\in U}\sigma(i,x)z_{i,h(i,x)}\left[(i,h(i,x))\in J\right]w_x\right\|_p
\end{aligned}$$

We can then use Equation (6) to get that

$$\begin{aligned}
&36e^3 \max_{J\subseteq[s]\times[m/s],|J|=k}\max_{\substack{z\in\mathbb{R}^{s\times m/s},\\ \|z\|_2=1}}\left\|\sum_{i\in[s]}\sum_{x\in U}\sigma(i,x)z_{i,h(i,x)}\left[(i,h(i,x))\in J\right]w_x\right\|_p \\
&\qquad\le 36e^3 \max_{J\subseteq[s]\times[m/s],|J|=k}\max_{\substack{z\in\mathbb{R}^{s\times m/s},\\ \|z\|_2=1}}\sqrt{L\frac{p}{\log m/s}}\,\|w\|_2\,\|z\|_2 \\
&\qquad= 36e^3\sqrt{L\frac{p}{\log m/s}}\,\|w\|_2
\end{aligned}$$

Combining the facts, we get that $\left\|\sum_{i\in[s]}\sum_{y\in U}\overline{\sigma}(i,y)v'_h(i,\overline{h}(i,y))w_y\right\|_p \le 36e^3 L\frac{p}{\log m/s}\|w\|_2^2$.

We will now bound $\left\|\sum_{i\in[s]}\sum_{y\in U}\overline{\sigma}(i,y)v''_h(i,\overline{h}(i,y))w_y\right\|_p$. We fix h and ε and use Equation (5) to get that

$$\begin{aligned}\left\|\sum_{i\in[s]}\sum_{y\in U}\overline{\sigma}(i,y)v''_h(i,\overline{h}(i,y))w_y\right\|_p &\leq \left\|\Psi_p\left(L\left\|w\right\|_\infty\left\|v'_h\right\|_\infty, L\frac{s}{m}\left\|w\right\|_2^2\left\|v'_h\right\|_2^2\right)\right\|_p \\ &\leq \left\|\Psi_p\left(L\left\|w\right\|_\infty\left|v'_h(\pi(k+1))\right|, L\frac{s}{m}\left\|w\right\|_2^2\left\|v_h\right\|_2^2\right)\right\|_p .\end{aligned}$$

Now we use Lemma 9 to get that,

$$\begin{aligned}&\left\|\Psi_p\left(L\left\|w\right\|_\infty\left|v'_h(\pi(k+1))\right|, L\frac{s}{m}\left\|w\right\|_2^2\left\|v_h\right\|_2^2\right)\right\|_p \\ &\qquad\leq \sqrt{2}\Psi_p\left(L\left\|w\right\|_\infty\left\|\left|v'_h(\pi(k+1))\right|\right\|_p, L\frac{s}{m}\left\|w\right\|_2^2\left\|\left\|v_h\right\|_2^2\right\|_{p/2}\right).\end{aligned}$$

Since we assume that $\frac{p}{\log m}\left\|w\right\|_2^2 < s\left\|w\right\|_2^2$ then Equation (7) give us that $\left\|\left\|v_h\right\|_2^2\right\|_{p/2} \leq Ls\left\|w\right\|_2^2$.

We will now bound $\left\|v'_h(\pi(k+1))\right\|_p$. For this, we will distinguish between two cases: Either $p \geq \log m$ or $p < \log m$. Let us first case where $p \geq \log m$. We will use that $\left|v'_h(\pi(k+1))\right| \leq \frac{\sum_{i\in[k+1]}\left|v'_h(\pi(i))\right|}{k+1}$. We then get that

$$\begin{aligned}&\left\|v'_h(\pi(k+1))\right\|_p \\ &\leq \left\|\frac{\sum_{i\in[k+1]}\left|v'_h(\pi(i))\right|}{k+1}\right\|_p \\ &\leq \left(\binom{m}{k+1}2^{k+1}\max_{\substack{J\subseteq[s]\times[m/s],\\ |J|=k+1}}\max_{(\sigma_{i,j})_{(i,j)\in J}\in\{-1,1\}^J}\left(\frac{\left\|\sum_{(i,j)\in J}\sigma_{i,j}v_h(i,j)\right\|_p}{k+1}\right)^p\right)^{1/p} \\ &\leq \max_{\substack{J\subseteq[s]\times[m/s],\\ |J|=k+1}}\max_{(s_{i,j})_{(i,j)\in J}\in\{-1,1\}^J} 2\binom{m}{k+1}^{1/p}\frac{\left\|\sum_{(i,j)\in J}\sigma_{i,j}v_h(i,j)\right\|_p}{k+1}\end{aligned}$$

We note that $\left\|\sum_{(i,j)\in J}\sigma_{i,j}v_h(i,j)\right\|_p = \left\|\sum_{x\in U}\sum_{(i,j)\in J}\sigma(i,x)s_{i,j}\left[h(i,x)=j\right]w_x\right\|_p$. Since we have that $p \geq \log m$ then $k \geq 1$ which implies that $k+1 \leq 2\frac{p}{\log(m/p)}$. We then get that $\binom{m}{k+1}^{1/p} \leq \left(\frac{em}{2p/\log(m/p)}\right)^{2/\log(m/p)} \leq 2e^3$. We now use Equation (6) to get that,

$$\begin{aligned}\left\|\sum_{x\in U}\sum_{(i,j)\in J}\sigma(i,x)s_{i,j}\left[h(i,x)=j\right]w_x\right\|_p &= \left\|\sum_{i\in[s]}\sum_{x\in U}\sigma(i,x)\left[(i,h(i,x))\in J\right]s_{i,h(i,x)}w_x\right\|_p \\ &\leq \sqrt{L\frac{p}{\log m/s}}\sqrt{|J|}\left\|w\right\|_2 \\ &= \sqrt{L\frac{p}{\log m/s}}\sqrt{k+1}\left\|w\right\|_2\end{aligned}$$

Combining this we get that $\left\| v_h'(\pi(k+1)) \right\|_p \le 4e^3 \sqrt{L \frac{p}{\log m/s}} \frac{\|w\|_2}{\sqrt{k+1}}$. We then obtain that,

$$\begin{aligned}
&\left\| \sum_{i \in [s]} \sum_{y \in U} \overline{\sigma}(i,y) v_h''(i, \overline{h}(i,y)) w_y \right\|_p \\
&\qquad \le \sqrt{2}\Psi_p \left(4e^3 L \sqrt{L \frac{p}{(k+1)\log m/s}} \left\| w \right\|_\infty \left\| w \right\|_2 , L^2 \frac{s^2}{m} \left\| w \right\|_2^4 \right) \\
&\qquad \le \sqrt{2}\Psi_p \left(4e^3 L \sqrt{L \frac{\log m/p}{\log m/s}} \left\| w \right\|_2^2 , L^2 \frac{s^2}{m} \left\| w \right\|_2^4 \right)
\end{aligned}$$

If $\log m/p \le 4 \log m/s$ then we get that,

$$\begin{aligned}
\left\| \sum_{i \in [s]} \sum_{y \in U} \overline{\sigma}(i,y) v_h''(i, \overline{h}(i,y)) w_y \right\|_p &\le \sqrt{2}\Psi_p \left(16e^3 L^{3/2} \left\| w \right\|_2^2 , L^2 \frac{s^2}{m} \left\| w \right\|_2^4 \right) \\
&\le \Psi_p \left(32e^3 L^{3/2} \left\| w \right\|_2^2 , 2L^2 \frac{s^2}{m} \left\| w \right\|_2^4 \right)
\end{aligned}$$

If $\log m/p > 4 \log m/s$ then $m/p > (m/s)^4$ which implies that $\frac{pm}{s^2} \le \sqrt{pm}$. Using this we get that $\frac{p 16 e^6 L^3 \frac{\log m/p}{\log m/s} \|w\|_2^4}{L^2 \frac{s^2}{m} \|w\|_2^4} \le 16e^6 L \sqrt{pm} \log m/p \le 16e^6 L$. Where we have used that $\sqrt{s} \log 1/x \le 1$. Now we use Lemma 10 to get that

$$\begin{aligned}
\left\| \sum_{i \in [s]} \sum_{y \in U} \overline{\sigma}(i,y) v_h''(i, \overline{h}(i,y)) w_y \right\|_p &\le \sqrt{2}\Psi_p \left(32e^3 L^{3/2} \left\| w \right\|_2^2 , L^2 \frac{s^2}{m} \left\| w \right\|_2^4 \right) \\
&\le \sqrt{2} \sqrt{p L^3 16 e^6 \frac{s^2}{m} \left\| w \right\|_2^4} \\
&\le \Psi_p \left(8e^3 L^{3/2} \left\| w \right\|_2 , 32 e^6 L^3 \frac{s^2}{m} \left\| w \right\|_2^4 \right)
\end{aligned}$$

Now let us consider the case where $p < \log m$. By Equation (8), we get that

$$\left\| v_h'(\pi(k+1)) \right\|_p \le \left\| \max_{i \in [s], j \in [m/s]} \left| \sum_{x \in U} \sigma(i,x) \left[h(i,x) = j \right] w_x \right| \right\|_p \le e \sqrt{L \frac{\log m}{\log m/s}} \left\| w \right\|_2$$

We then obtain that,

$$\begin{aligned}
\left\| \sum_{i \in [s]} \sum_{y \in U} \overline{\sigma}(i,y) v_h''(i, \overline{h}(i,y)) w_y \right\|_p &\le \sqrt{2}\Psi_p \left(eL \sqrt{L \frac{\log m}{\log m/s}} \left\| w \right\|_\infty \left\| w \right\|_2 , L \frac{s^2}{m} \left\| w \right\|_2^4 \right) \\
&\le \sqrt{2}\Psi_p \left(eL \sqrt{L \frac{\log m}{\log m/s}} \left\| w \right\|_2^2 , L \frac{s^2}{m} \left\| w \right\|_2^4 \right)
\end{aligned}$$

If $s \le m^{3/4}$ then we get that $\log m \le 4 \log m/s$ and

$$\sqrt{2}\Psi_p \left(eL \sqrt{L \frac{\log m}{\log m/s}} \left\| w \right\|_2^2 , L \frac{s^2}{m} \left\| w \right\|_2^4 \right) \le \sqrt{2}\Psi_p \left(4eL^{3/2} \left\| w \right\|_2^2 , L \frac{s^2}{m} \left\| w \right\|_2^4 \right)$$

as wanted. If $s \geq m^{3/4}$ then we get that $\frac{peL^3 \log m}{L^2 s^2/m} \leq \frac{eLm \log^2 m}{s^2} \leq \frac{eL \log^2 m}{m^{1/2}} \leq 16/eL$, where we have used that $\sqrt{x} \log^2 1/x \leq 16/e^2$. Again we use Lemma 10 to get that

$$\begin{aligned}\sqrt{2}\Psi_p\left(4eL^{3/2}\left\|w\right\|_2^2, L\frac{s^2}{m}\left\|w\right\|_2^4\right) &\leq \sqrt{p32/eL^3\frac{s^2}{m}}\left\|w\right\|_2^2 \\ &\leq \Psi_p\left(8L^{3/2}\left\|w\right\|_2^2, 32L^3\frac{s^2}{m}\left\|w\right\|_2^4\right).\end{aligned}$$

Combining everything we get that

$$\begin{aligned}\left\|\sum_{i\in[s]}\sum_{y\in U}\overline{\sigma}(i,y)v_h''(i,\overline{h}(i,y))w_y\right\|_p &\leq \Psi_p\left(32e^3L^{3/2}\left\|w\right\|_2, 32e^6L^3\frac{s^2}{m}\left\|w\right\|_2^4\right), \\ \left\|\sum_{i\in[s]}\sum_{y\in U}\overline{\sigma}(i,y)v_h'(i,\overline{h}(i,y))w_y\right\|_p &\leq 4e^3L\frac{p}{\log m/s}\left\|w\right\|_2^2.\end{aligned}$$

Now we conclude that

$$\begin{aligned}&\left\|\sum_{i\in[s],j\in[m/s]} v_h(i,j)v_{\overline{h}}(i,j)\right\|_p \\ &\quad\leq \left\|\sum_{i\in[s]}\sum_{y\in U}\overline{\sigma}(i,y)v_h'(i,\overline{h}(i,y))w_y\right\|_p + \left\|\sum_{i\in[s]}\sum_{y\in U}\overline{\sigma}(i,y)v_h''(i,\overline{h}(i,y))w_y\right\|_p \\ &\quad\leq \Psi_p\left(32e^3L^{3/2}\left\|w\right\|_2^2, 32e^6L^3\left\|w\right\|_2^4\right) + 4e^3L\frac{p}{\log m/s}\left\|w\right\|_2^2.\end{aligned}$$

Thus finishing the proof. ◀

References

1 Anders Aamand, Jakob Bæk Tejs Knudsen, Mathias Bæk Tejs Knudsen, Peter Michael Reichstein Rasmussen, and Mikkel Thorup. Fast hashing with strong concentration bounds. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pages 1265–1278, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3357713.3384259.

2 Anders Aamand and Mikkel Thorup. Non-empty bins with simple tabulation hashing. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2498–2512. SIAM, 2019. doi:10.1137/1.9781611975482.153.

3 Dimitris Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003. Special Issue on PODS 2001. doi:10.1016/S0022-0000(03)00025-4.

4 Nir Ailon and Bernard Chazelle. The fast johnson–lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009. doi:10.1137/060673096.

5 Nir Ailon and Edo Liberty. Fast dimension reduction using rademacher series on dual BCH codes. In Shang-Hua Teng, editor, *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*, pages 1–9. SIAM, 2008. URL: http://dl.acm.org/citation.cfm?id=1347082.1347083.

6 Noga Alon and Bo'az Klartag. Optimal compression of approximate inner products and dimension reduction. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 639–650, October 2017. doi:10.1109/FOCS.2017.65.

7 Stefan Bamberger and Felix Krahmer. Optimal fast johnson-lindenstrauss embeddings for large data sets. *Sampling Theory, Signal Processing, and Data Analysis*, 19, June 2021. doi:10.1007/s43670-021-00003-5.

8 Vladimir Braverman, Rafail Ostrovsky, and Yuval Rabani. Rademacher chaos, random eulerian graphs and the sparse johnson-lindenstrauss transform. *CoRR*, abs/1011.2590, 2010. arXiv:1011.2590.

9 Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theoretical Computer Science*, 312(1):3–15, 2004. Automata, Languages and Programming. doi:10.1016/S0304-3975(03)00400-6.

10 Michael B. Cohen, T. S. Jayram, and Jelani Nelson. Simple analyses of the sparse johnson-lindenstrauss transform. In Raimund Seidel, editor, *1st Symposium on Simplicity in Algorithms, SOSA 2018, January 7-10, 2018, New Orleans, LA, USA*, volume 61 of *OASIcs*, pages 15:1–15:9. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/OASIcs.SOSA.2018.15.

11 S. Dahlgaard, M. B. T. Knudsen, E. Rotenberg, and M. Thorup. Hashing for statistics over k-partitions. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1292–1310, 2015. doi:10.1109/FOCS.2015.83.

12 Søren Dahlgaard, Mathias Bæk Tejs Knudsen, and Mikkel Thorup. Practical hash functions for similarity estimation and dimensionality reduction. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6618–6628, USA, 2017. Curran Associates Inc. URL: http://dl.acm.org/citation.cfm?id=3295222.3295407.

13 Søren Dahlgaard and Mikkel Thorup. Approximately minwise independence with twisted tabulation. In R. Ravi and Inge Li Gørtz, editors, *Algorithm Theory – SWAT 2014*, pages 134–145, Cham, 2014. Springer International Publishing.

14 Anirban Dasgupta, Ravi Kumar, and Tamás Sarlos. A sparse johnson: Lindenstrauss transform. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10, pages 341–350, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1806689.1806737.

15 Thong T. Do, Lu Gan, Yi Chen, Nam Nguyen, and Trac D. Tran. Fast and efficient dimensionality reduction using structurally random matrices. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1821–1824, 2009. doi:10.1109/ICASSP.2009.4959960.

16 Ora Nova Fandina, Mikael Møller Høgsgaard, and Kasper Green Larsen. Barriers for faster dimensionality reduction, 2022. doi:10.48550/arXiv.2207.03304.

17 Casper Freksen, Lior Kamma, and Kasper Green Larsen. Fully understanding the hashing trick. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 5394–5404, Red Hook, NY, USA, 2018. Curran Associates Inc.

18 Casper Benjamin Freksen and Kasper Green Larsen. On using toeplitz and circulant matrices for johnson-lindenstrauss transforms. *Algorithmica*, 82(2):338–354, 2020. doi:10.1007/s00453-019-00644-y.

19 Aicke Hinrichs and Jan Vybíral. Johnson-lindenstrauss lemma for circulant matrices. *Random Structures & Algorithms*, 39(3):391–398, 2011. doi:10.1002/rsa.20360.

20 Jakob Bæk Tejs Houen and Mikkel Thorup. Understanding the moments of tabulation hashing via chaoses. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 74:1–74:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.74.

21 Meena Jagadeesan. Understanding sparse jl for feature hashing. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, NeurIPS'19, Red Hook, NY, USA, 2019. Curran Associates Inc.

22 Vishesh Jain, Natesh S. Pillai, Ashwin Sah, Mehtaab Sawhney, and Aaron Smith. Fast and memory-optimal dimension reduction using Kac's walk. *The Annals of Applied Probability*, 32(5):4038–4064, 2022. doi:10.1214/22-AAP1784.

23 T. S. Jayram and David P. Woodruff. Optimal bounds for johnson-lindenstrauss transforms and streaming problems with subconstant error. *ACM Trans. Algorithms*, 9(3), June 2013. doi:10.1145/2483699.2483706.

24 William Johnson and Joram Lindenstrauss. Extensions of lipschitz maps into a hilbert space. *Contemporary Mathematics*, 26:189–206, January 1984. doi:10.1090/conm/026/737400.

25 Mark Kac. Foundations of kinetic theory. In *Proceedings of The third Berkeley symposium on mathematical statistics and probability*, volume 3, pages 171–197, 1956.

26 Daniel Kane, Raghu Meka, and Jelani Nelson. Almost optimal explicit johnson-lindenstrauss families. In Leslie Ann Goldberg, Klaus Jansen, R. Ravi, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 628–639, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

27 Daniel M. Kane and Jelani Nelson. A derandomized sparse johnson-lindenstrauss transform, 2010. doi:10.48550/arXiv.1006.3585.

28 Daniel M. Kane and Jelani Nelson. Sparser johnson-lindenstrauss transforms. *J. ACM*, 61(1), January 2014. doi:10.1145/2559902.

29 Felix Krahmer and Rachel Ward. New and improved johnson–lindenstrauss embeddings via the restricted isometry property. *SIAM Journal on Mathematical Analysis*, 43(3):1269–1281, 2011. doi:10.1137/100810447.

30 Kasper Green Larsen and Jelani Nelson. Optimality of the johnson-lindenstrauss lemma. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 633–638, 2017. doi:10.1109/FOCS.2017.64.

31 Jelani Nelson and Huy L. NguyÅn. Sparsity lower bounds for dimensionality reducing maps. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 101–110, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2488608.2488622.

32 Mihai Patrascu and Mikkel Thorup. Twisted tabulation hashing. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 209–228. SIAM, 2013. doi:10.1137/1.9781611973105.16.

33 Mihai Pătraşcu and Mikkel Thorup. The power of simple tabulation hashing. *J. ACM*, 59(3), June 2012. doi:10.1145/2220357.2220361.

34 Alan Siegel. On universal classes of extremely random constant-time hash functions. *SIAM Journal on Computing*, 33(3):505–543, 2004. Announced at FOCS'89.

35 Mikkel Thorup. Simple tabulation, fast expanders, double tabulation, and high independence. In *54th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 90–99, 2013.

36 Mikkel Thorup and Yin Zhang. Tabulation-based 5-independent hashing with applications to linear probing and second moment estimation. *SIAM Journal on Computing*, 41(2):293–331, 2012. doi:10.1137/100800774.

37 Jan Vybíral. A variant of the johnson–lindenstrauss lemma for circulant matrices. *Journal of Functional Analysis*, 260(4):1096–1105, 2011. doi:10.1016/j.jfa.2010.11.014.

38 Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1113–1120, New York, NY, USA, 2009. Association for Computing Machinery. doi:10.1145/1553374.1553516.

39 Albert Lindsey Zobrist. A new hashing method with application for game playing. Technical Report 88, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, 1970.